AutoMelter

Youssef Abdelhalim
under supervision of Ph.D. Candidate Hebert Azevedo-Sa

Table of contents:

What is it?

AutoMelter is a device that melts snow off of driveways and roads automatically by releasing a solution of ethanol, sodium stearate, sodium chloride, and water. Here's how it works:

1. A temperature sensor and a precipitate sensor work simultaneously (functioning as a snow sensor when they work in conjunction) to sense the presence of snow.
2. If there is snow, a signal is sent to an electric valve that is connected to the tank that contains the solution.
3. The electric valve is connected to a pipe that has holes in it, allowing the solution to sprout out of the sides of the pipe onto the surface.
4. The solution is heated and has a super low freezing point, this makes it impossible for snow to stick to the surface and could melt snow that has already been piled up

This is the basic idea of the device, however, it goes much deeper than that.

Why is AutoMelter important?

There are two main reasons this invention is extremely convenient and efficient.

The first and most important one is public safety. Gritters (salt trucks) are the main means of clearing snow off the streets around the US and the rest of the world as well. However, they are dangerous and inefficient. Every year, around 2,000 deaths and 140,000 injuries are attributed to icy or snowy roads, according to the US Department of Transportation. In fact, salt trucks themselves could lose control and get into fatal crashes. Last year, a salt truck lost control and flipped over, killing both the driver and the operator in a neighboring town. Salt trucks were invented in 1942 and their designs were not updated ever since.

Apart from the fact that they're dangerous, they're also extremely inefficient. A salt truck costs around $100,000 (may vary depending on the size and machinery included in the truck), however, they are only used seasonally, and sometimes, they're not used at all if a winter is not too snowy- such as the winter of 2019, where it snowed twice in my local area and both times it was just a couple of inches. They also release a lot more Carbon Monoxide compared to cars, which is harmful to the environment, atmosphere, and people.

Moreover, there have been thousands of people reporting getting cracks on their windshields and scratches all over their cars' bodies due to the thick rock salt particles that get released from these salt trucks.

AutoMelter would eliminate the need for gritters. This would save more lives, make roads safer since they work instantly as soon as it starts snowing, and save a lot of money needed for the manufacturing and operating of salt trucks.

The other reason AutoMelter is important is the general public's convenience. Shoveling snow is extremely tedious and inconvenient, especially when there is two feet of snow piled up all along the driveway. AutoMelter can solve this issue by again, preventing snow from even sticking in the first place.

This was actually one of the reasons I started working on this project. Since I am the youngest in my family, I have always been the one that had to shovel the snow off the driveway and I despised it. I started looking for ways to decrease the time I spent shoveling snow, and I was met with three solutions, all of which didn't seem too efficient or cheap:

1. Just shoveling: physically demanding and time-consuming.
2. Getting an electric snow plow: expensive (avg. cost ~ $3,000) and would still require me to go outside to plow the snow.
3. Getting a heated driveway: extremely expensive (cost ranges between $7,000-$25,000 depending on the material of the driveway and the size of the driveway), can only be installed when a house is being built otherwise one would have to pay construction workers to break his/her driveway and redo it, and maintenance would be extremely expensive and laborious since it would require digging in the ground.

After exploring all those options, I realized that there wasn't one cheap or effective solution on the market. So I went to the drawing board and designed AutoMelter that tackles every single one of those problems: it is cheap to assemble, it is automatic so one wouldn't need to go out at all to get rid of the snow on the driveway, and it is extremely easy to repair and adjust based on an individual's liking.

How does it work?

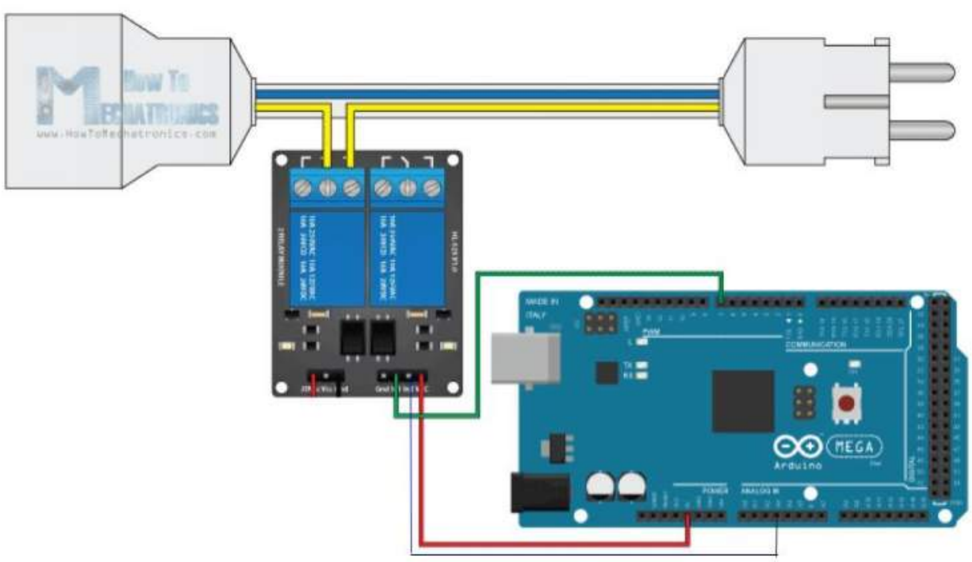
AutoMelter consists of four systems which I will be explaining in detail (I will be including pictures and videos too) for anyone who wishes to recreate it. They are: electronics, coding, building, and RemoteXY (mobile app)

- Electronics

We will start with electronics. (This will require all the products under "electronics" in the products list.) The electronics could be divided into 3 sub-systems for simplicity's sake:

- water-heating system,
- valve-controlling system, and
- RemoteXY connection.

The water heating system requires the water boiler, temperature sensor, relay, Arduino, connection cables for Arduino, and copper wires.



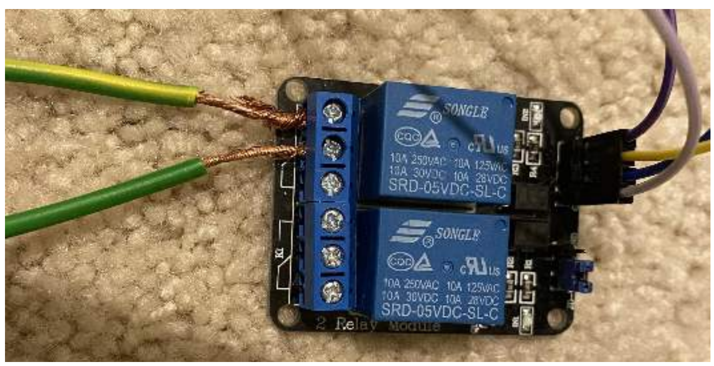
(Image from Arduino Relay Tutorial - Control High Voltage Devices with Arduino)

This is what the system output of the system should look like. Now I will show you step-by-step how to get there:

1. Connect two copper wires to each end of the water boiler's plug. Make sure you loop it around the hole (like the blue wire) to ensure that it tightly fits. Then wrap it with electric tape so every part is covered (like the green wire)

2. Then for the other end of the copper wires, connect them to small metal pieces (I used the metal pieces from an old plug) and wrap them around the metal pieces tightly to make sure that the current flows through. Then wrap it with electric tape.
3. After the first two steps, you should have this:

4. Then, find the middle of the green or blue wire (it doesn't matter which one) and cut it. Then cut a little piece of insulation to expose the wire itself. Plug in each end to the two left ports of K2 on the relay.



5. Connect 4 female-to-male wires to the other side of the relay board as seen.
    a. GND to a GND outlet on the breadboard
    b. IN2 to Digital 3 on Arduino
    c. VCC to a 5V outlet on the breadboard
6. Now, we are done with the output. I will show you how to connect the input (temperature sensor)
7. Get your temperature sensor and the connection board that comes with it and connect it in the following configuration:
    a. Black - GND
    b. Red - VCC
    c. Yellow - DAT

8. For the other ends (using the same colors preferably), connect them as follows:
    a. Black - to a GND outlet on the breadboard
    b. Red - to a 5V outlet on the breadboard
    c. Yellow - in an outlet on the breadboard that connects to Digital 5

Now we will start on the second electronics sub-system, the one involved with Controlling the valve:

1. Get your temperature sensor and the connection board that comes with it and connect it in the following configuration:
    a. Black - GND
    b. Red - VCC
    c. Yellow - DAT

2. For the other ends (using the same colors preferably), connect them as follows:
    a. Black - to a GND outlet on the breadboard
    b. Red - to a 5V outlet on the breadboard
    c. Yellow - to a Digital 5 outlet on the breadboard
3. Get your precipitate sensor and the connection board that comes with it and connect it in the following configuration:

4. For the other ends, connect them as follows:
    a. VCC to a 5V outlet on the breadboard
    b. GND to a GND outlet on the breadboard
    c. D0 to Digital 4 on Arduino
    d. A0 to Analogue 0 on Arduino
5. Get two pieces of copper wires and connect them to the valve as shown (the same way they were connected to the water boiler). Make sure you wrap them with electrical tape

6. Then connect the other ends to the connector that came with the power adaptor.

7. Take one of the cables (I chose the green one) and cut it through the middle, just like we did with the water boiler. Then connect the two new

ends that were formed and connect them to the following slots in the relay (K1). They're the ones to the right:

8. Connect In2 on the other side of the relay module to Digital 6 on the Arduino using a female-to-male connection wire

Now we'll connect our ESP8266 to the Arduino. This allows us to connect the RemoteXY app from our phone to the Arduino. By doing this we override the sensors' information and control the water-boiler system and the valve system manually.

You will need to upload your code first before connecting to this system. If you try to connect it first and then upload the code, the code will not upload to the Arduino. This is because the signals from the ESP8266 will interfere with the Arduino's connection. To get around this, just upload the code first, then connect ESP8266 to the Arduino

1. Get a female-to-male cable, cut it from the middle, and then shave off the insulator ends. Get another cable and cut the female part, shave off the insulator end, and connect all three of the new ends so it looks like that:

2. Then get your ESP8266 and connect it as follows:
    a. RX - to the TX port on the Arduino (Digital 1)
    b. TX - to the RX port on the Arduino (Digital 0)
    c. GND - to a GND port on the Arduino or the breadboard
    d. 3V3 + EN - to the 3.3 V port on the Arduino (use the connection from part 1 to achieve this connection)

- Coding

For this section, I will be posting snippets of the code and explaining what each one means. Most of the variables could be set based on preference so you could customize your device based on the snow levels you are expecting.

1. First, we need to add some libraries and define some variables. we need to assign the pins to their respective variables and set a minimum and maximum temperature for the water heating system, this could be set based on preference, however, after thorough testing, 0 to 20 has proved to be the optimum range. I also defined sensor 1 and sensor 2 to prevent confusion when we are writing the code for each system.

```
#define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT

#include <RemoteXY.h>

#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 115200
#define REMOTEXY_WIFI_SSID "RemoteXY"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 6377

#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
  { 255,1,0,0,0,20,0,11,13,0,
  2,0,37,22,22,11,2,26,31,31,
  79,78,0,79,70,70,0 };

struct {
  uint8_t switch_1;
  uint8_t connect_flag;
}

RemoteXY;
#pragma pack(pop)

#define PIN_SWITCH_1 13

#include <OneWire.h>
#include <DallasTemperature.h>
#define RELAYVALVE 6
#define RELAYBOILER 3
#define ONE_WIRE_BUS 5

// Sensor 1 = water temp
// Sensor 2 = air temp
```

```
const int capteur_D = 4;
int deviceCount = 0;
float tempC;

float tempWater;
float tempAir;

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

double minTemp = 0, maxTemp = 20;
double startTime, currTime;

bool senseWater(){
  return !digitalRead(capteur_D);
```

2. I then wrote my setup function

```
void setup() {
  RemoteXY_Init ();
  pinMode(RELAYVALVE, OUTPUT);
  pinMode(RELAYBOILER, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(capteur_D, INPUT);
  sensors.begin();
  deviceCount = sensors.getDeviceCount();
  Serial.print(deviceCount, DEC);
  pinMode (PIN_SWITCH_1, OUTPUT);
}
```

3. Then comes the loop function. Since the loop function is more complicated and longer, I will be dividing it into 3 sections.
    a. The first part of my loop function has the lines of code that print out the values received from the sensors and the RemoteXY app onto the serial monitor. This makes it easier to keep track of the values and make sure that everything is working correctly and that the sensors are calibrated.

```
void loop() {
  RemoteXY_Handler ();
  sensors.requestTemperatures();
  for (int i = 0;  i < deviceCount;  i++)
  {
  Serial.print("Sensor ");
  Serial.print(i+1);
  Serial.print(" : ");
  tempC = sensors.getTempCByIndex(i);
  Serial.print(tempC);
  Serial.print((char)176);
  Serial.print("C  |  ");
  Serial.print(DallasTemperature::toFahrenheit(tempC));
  Serial.print((char)176);
  Serial.println("F");
  }
  Serial.print("Water: ");
  Serial.println(senseWater());
  delay(1000);

  tempWater = sensors.getTempCByIndex(0);
  tempAir = sensors.getTempCByIndex(1);
```

    b. The following part is the code for the water heating system. The logic means that when the temperature sensor reads that the temperature of the water is lower than the minimum temperature (0 ºC) or when the switch from RemoteXY is turned on, it will trigger the relay to close the circuit to turn on the boiler to heat the water. And when it reads that the water temperature is higher than the maximum temperature (20 ºC) or when the switch from RemoteXY is turned off, it triggers the relay to open the circuit to turn off the boiler so the water does not become too warm.

```
if(tempWater < minTemp || RemoteXY.switch_1==1)
 {
   delay(10);
   digitalWrite(RELAYBOILER, LOW);
   delay(10);
 }
 else if(tempWater > maxTemp)
 {
   delay(10);
   digitalWrite(RELAYBOILER, HIGH);
   delay(10);
 }
```

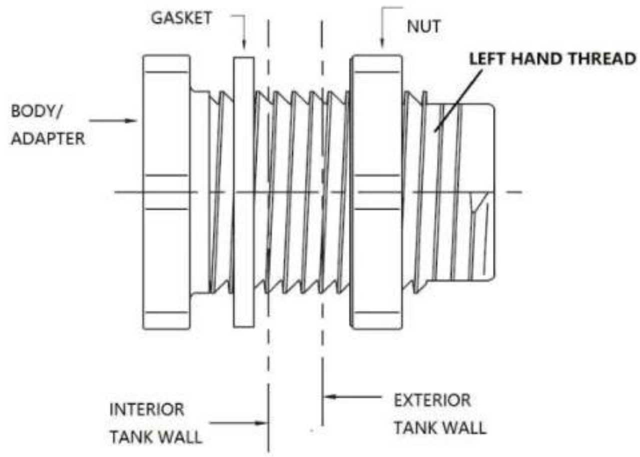
c. The following snippet is the piece of code associated with the valve-controlling system. This part consists of an if-else statement and another if-else statement nested within the main if statement. The logic of the main if-else statement is to let the valve open when the sensors sense the presence of snow (when the temp sensor reads less than 0 ºC and the precipitate sensor reads “wet”) or when the switch on the RemoteXY app is turned on and close the valve when one of these conditions, or either of them is not met, or when the switch from RemoteXY is turned off.
The nested if-else statement is there so that the valve is not opened all the time whenever these two conditions are met. This is to ensure that the tank doesn’t empty within the first few hours of a longer snowstorm (12+ hrs). This allows the valve to open for a certain amount of time every hour and closed for the rest, even if the sensors do sense snow.
I wrote my code such that the valve is opened for 10 minutes every hour and closed for the rest of the 50 minutes even if there is still snow unless the switch from RemoteXY is turned on. In that case, the timer doesn’t have an effect and the valve will stay open until the switch in RemoteXY is off.

```
 if(senseWater() && tempAir <= 25 || RemoteXY.switch_1==1)
{
 currTime = millis();
 if((currTime - startTime) > 10 * 60 * 1000)
   startTime = currTime;
 if((currTime - startTime) <= 50 * 60 * 1000){
   delay(10);
   digitalWrite(LED_BUILTIN, HIGH);
   digitalWrite(RELAYVALVE, LOW);
   delay(10);
 }
 else
 {
   delay(10);
   digitalWrite(LED_BUILTIN, LOW);
   digitalWrite(RELAYVALVE, HIGH);
   delay(10);
 }
 }
 else
 {
   delay(10);
   digitalWrite(LED_BUILTIN, LOW);
   digitalWrite(RELAYVALVE, HIGH);
   delay(10);
 startTime = millis();
 }
}
```

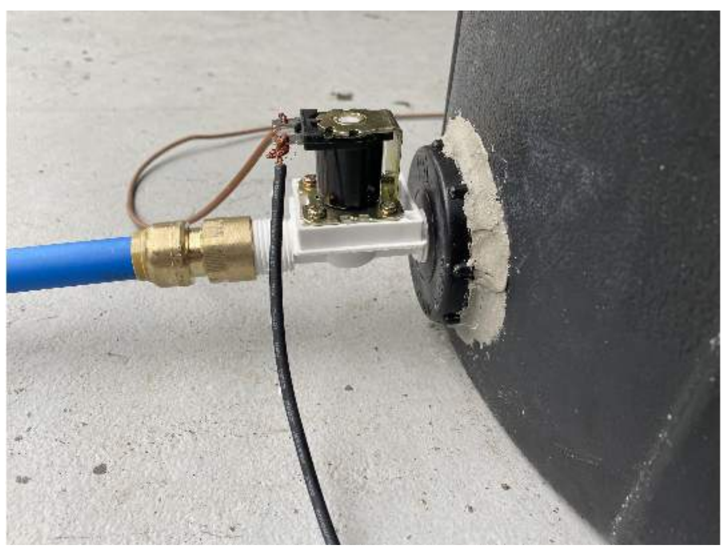
- **Build**

This part should be fairly easier than the first two parts. Get all your hardware and follow the steps.

1. Get your trash can and drill a hole using your drill and 1 ½ inch spade bit towards the bottom, as low as possible to ensure that the water has the greatest potential energy when it is leaving the valve. Make sure you go in with the drill gently and slowly to prevent any cracks from forming around the hole.
2. After you drill the hole, use a knife or sandpaper to make the hole nice and clean from all the shavings
3. Then insert the ½” everbilt bulkhead union with EPDM washer female.
    a. Take the nut off and insert the bulkhead from the inside such that the gasket is tight against the interior wall. Put the nut and screw it on from the outside such that it is tight against the outside wall. Don’t over-tighten the bulkhead from the inside or the nut from the outside because that can cause minor cracks as well. Make sure they’re tight but don’t overdo it.



b. After that, apply some sealant on the inside and outside around the nut and the gasket. This step is not needed but highly recommended. This is because not all trash cans are made using the same material, this means that there could be micro cracks around the hole, which will let some water through. The sealant prevents water leaks at high water pressures too.



4. Then use the ½ pipe thread tape around both ends of the valve. Make sure the valve is nice and tight to prevent any water leaks. Make sure your valve is pointing in the direction of the water flow. These valves are unidirectional, meaning the water could only flow in only one direction. You can figure out the direction by looking at the top of the valve. There will be an arrow that points in the direction of the water flow.
5. Then connect the other end of the valve to the ½" x ½" sharkbite FNPT female adapter.

6. Now get one of the three pipes and connect it to your ½" x ½" sharkbite FNPT female adapter. You can cut your pipe down to any preferred length or leave it as is. If you would like to have your tank further inside your garage then leave it as is, but if you want the tank closer to the door, then cut the pipe accordingly.

7. To the other end of that pipe, connect the ½" x ½" x ½" sharkbite T junction connection.

8. Then connect the other two pipes to either end of the ½" x ½" x ½" sharkbite T junction connection. Make sure you cut your pipes evenly to fit the garage width. You want to cut them evenly so the water pressure is constant in both tubes.
9. Add the ½" sharkbite end stop to the ends of both the tubes as such

10. Then drill holes using the Ryobi speedload+ titanium 4 piece hex shank pilot drill set. This part is tricky so make sure you're being super careful.
    a. Make sure all the holes are aligned in a straight line
    b. Make sure all the holes are evenly spaced
    c. Make sure all the holed are the same size
    d. I used the 3/32 inch bit and had 6 holes in each tube spaced out evenly. But you can choose whichever size you prefer and however many holes you prefer. This is because not all tanks are the same size, which means the water pressure in my tank might not be the same as yours, so adjust your pipes accordingly to make sure that the pressure of the water coming out of the tubes is set to your liking.
    e. Make sure you start off with the smallest size drill bit because you can always go bigger and widen the hole but you can't go smaller.
    f. Make sure you start off with a few holes and increase them based on your liking because again, you can easily drill more holes but it is a hassle to seal existing holes.

(Here's how big my holes were just so you're able to visualize how wide 3/32 inch is).
11. Submerge the water boiler into the tank, but make sure the ending where the plug is connected to the blue and green cables is not submerged in. You can tape it to the side of the container to prevent it from falling in. Also, make sure the boiler is always submerged whenever it is connected. This is because it will melt if it is not in water.
12. Your solution will consist of 3 fluids. The following measurements are variable depending on the size of your tank. Mine is 32 gallons. If yours isn't, just scale the quantities up or down proportionally.
    a. 30-31 gallons of water
    b. 1.5 gallons of alcohol (ethanol)

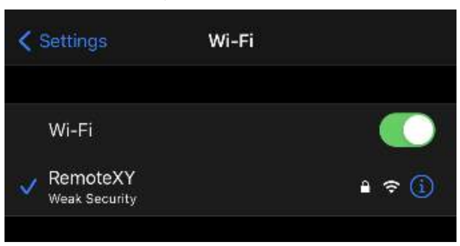
c. 3 ounces of dish soap (sodium stearate)
d. Ratio= ½ gallon water: 6 drops of dish soap: ¼ cup of alcohol

- RemoteXY

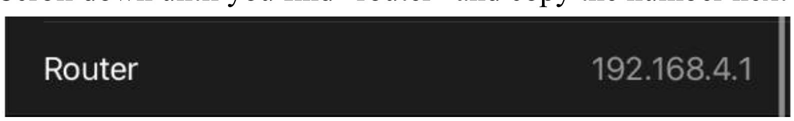
This part explains how to control the system through your phone in case you prefer to have the system constantly turned on rather than switching depending on time intervals. My explanation will be in terms of IOS, but if you have an Android I will make sure I leave a video explaining how to use it on any Android at the end of this document.

1. Go to the app store and download "RemoteXY" developed by Evgeny Shemanuev.

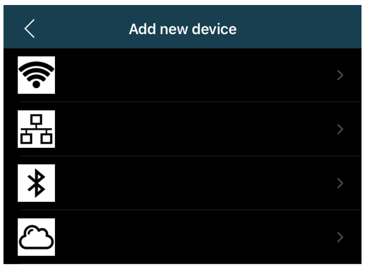
2. Upload the code to the Arduino before connecting ESP8266EX.
3. After you upload the code, connect ESP8266EX, then go to your wifi settings, and connect your wifi to "RemoteXY".



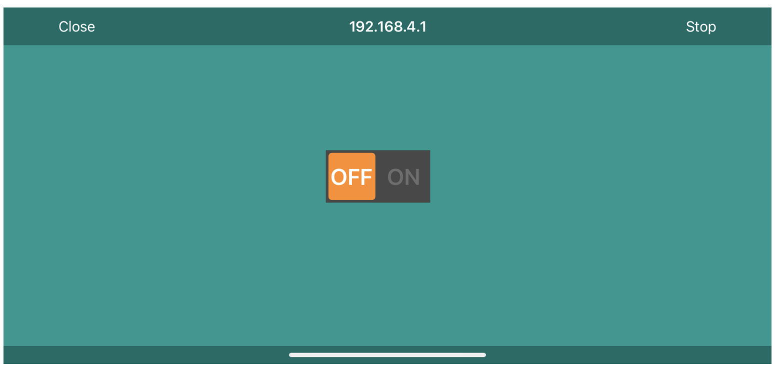
4. Then click on the "i" button to bring up more information about the connection.
5. Scroll down until you find "router" and copy the number next to it.



6. Then go to your RemoteXY app, click on the "+" sign in the top right corner, then select the second option, then click on "IP or URL" and paste your router IP address and click connect.



7. Then a screen with an on/off switch will pop up.



Materials for the build	Prices (per piece)
Hardware:	
Container (a trash can works)	38.99
3x ½” x 5’ pipe	5.58
2x ½” sharkbite end stop	12.40
½” sharkbite disconnect clip	1.86
½” x ½” x ½” sharkbite T junction connection	11.52
½” x ½” sharkbite FNPT female adapter	6.85
HDX 25 ft indoor/outdoor extension cord	10.97
½” everbilt bulkhead union with EPDM washer female	14.45
Irwin speedbor 1 ½” spade bit	7.18
Craftsman V20 drill	79.99
Ryobi speedload+ titanium 4 piece hex shank pilot drill set	4.97
½” pipe tape tape	0.59
Dynaflex Sealant	4.68
Electronics:	
Arduino Uno	23.00
Breadboard, Female-female connection cables, Male-male connection cables, Female-male connection cables	14.99
Power adapter (pack of 2)	9.99
12V Water Solenoid Valve	10.29
Precipitate Sensor	4.98
2x Temperature sensor	17.98
5V 2-channel relay module	7.99

Water Boiler	7.99
Copper wire	9.16
Electrical tape	1.99
ESP8266 EX (Pack of 5)	11.99
Total	320.38*

*201.40 without the trash can and drill since most people have that already

Datasheets:

1. Arduino Uno
2. 12V Water Solenoid Valve
3. Precipitate Sensor
4. Temperature Sensor
5. Relay
6. ESP8266EX

Helpful links:

1. Arduino Relay Tutorial - Control High Voltage Devices with Arduino
2. Arduino Solenoid Valve Circuit: How to Control Water Flow with an Arduino
3. Interfacing Multiple DS18B20 Digital Temperature Sensors with Arduino
4. Arduino Uno & ESP8266 and control using smartphone (Android)
    a. RemoteXY
5. How to Install A Bulkhead Fitting
6. How To Cut Pipe Straight/Square With A Hand Saw. Large Or Small! PVC, Polycarbonate, Acrylic, ECT
7. How to Cut PEX Pipe
8. How to drill aligned holes in a tube
9. Drawing a straight line on a round tube (only relevant until 2:40)
10. Solution proportions